UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/733,788	12/10/2003	Mehmet Yunt	MWS-106RCE2	8974

74321 7590 10/19/2009

LAHIVE & COCKFIELD, LLP/THE MATHWORKS
FLOOR 30, SUITE 3000
One Post Office Square
Boston, MA 02109-2127

EXAMINER	
SILVER, DAVID	

ART UNIT	PAPER NUMBER
2128	

MAIL DATE	DELIVERY MODE
10/19/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/733,788	YUNT ET AL.	
	Examiner	Art Unit	
	DAVID SILVER	2128	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 July 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9 and 11-72* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9 and 11-72* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The Instant Office Action is in response to a Request for Continued Examination filed 7/31/09.

2. Claims 1-9 and 11-72 are currently pending in Instant Application.

Response to Arguments

Response: 35 U.S.C. § 112

3. **Applicants argue:**

3.1 "In order to expedite prosecution, Applicants amend the independent claims to recite a graphical debugger that interfaces with ... Support for this feature of claim 1 can be found throughout the application as filed. For example, support can be found at page 5, last paragraph. Further support can be found at page 8, lines 10-20; at page 38, line 21 through page 39, line 15; and at Figure 15." (Remarks: page 1)

4. **Examiner Response:**

4.1 Applicants' amendments are sufficient to overcome the 35 U.S.C. § 112 P1 rejection. Accordingly, the rejection is withdrawn.

Response: 35 U.S.C. § 102 / 103

5. **Applicants argue:**

5.1 "MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest a block method execution list view of block methods called during execution of said model, nor do the cited references disclose or suggest said debug information indicating an order of execution of said plurality of block methods for said block and a start time or a stop time of said plurality of block methods for said block that are executed during the execution of said model, nor do the cited references disclose or suggest said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods that are executed in said block during the execution of said model as present in claim 1." (Remarks: page 2)

5.2 "A block represents a unit of functionality in a model (Specification at page 3, first full paragraph). Block methods provide a means of implementing the functionality of the block. In addition to block

methods, blocks may contain other information as well, such as numerical information (Specification at page 14, first full paragraph, and page 15, line 32, through page 17, line 4).

5.3 The Examiner asserts that the MathWorks reference discloses an execution list view at page 3-49 (Office Action at page 4). However, the cited passage of the MathWorks reference does not address block methods. Instead, the MathWorks reference explicitly states that the list view shown on page 3-49 depicts a list of blocks, and not block methods (MathWorks at page 3- 49, stating that "the blocks list" contains "the names of blocks in the selected system")." (Remarks: page 3)

6. **Examiner Response:**

6.1 Applicants' arguments have been fully considered but are not persuasive. Specifically, the blocks displayed in MathWorks necessarily must have methods, for those block to function. Otherwise, they were merely be ironical representations without functionality. For those blocks to be functional, they must have execution methods "behind the scenes" and functionally contained therein. MathWorks indeed discloses a list of blocks, and therefore a list of block methods, as said blocks contain block methods. This is consistent with Applicants' own disclosure (pointed to by Applicants in the arguments recited above), and is considered prior-art as it is disclosed in the Background section of the Specification: "Elements of a block diagram such as blocks, subsystems and the underlying model contain a collection of methods that are invoked by the execution engine at certain times during the simulation for different purposes, e.g., to generate an output, to update the state, and to compute derivatives." (page 3, first full paragraph) Accordingly, the position is maintained.

7. **Applicants argue:**

7.1 On page 3 to page 5 Applicants argue that Alpern does not remedy the above-alleged deficiency. Applicants then conclude that the Official Notice taken also does not remedy this alleged deficiency.

8. **Examiner Response:**

8.1 As demonstrated above, Applicants' position was been traversed. Accordingly, these arguments are moot.

9. **Applicants argue:**

9.1 "In light of the above discussion, the cited references also do not disclose or suggest said debug information indicating an order of execution of said plurality of block methods for said block and a start time or a stop time of said plurality of block methods for said block that are executed during the execution of said model. The debugger in the MathWorks reference provides debug information indicating information about the blocks, and does not indicate an order of execution of said plurality of block methods for said block nor a start time or a stop time of said plurality of block methods. Alpern may provide debug information about methods called during the execution of generated code, but the code of Alpern does not correspond to block methods for the reasons discussed above. The Official Notice taken does not address debug information at all.

9.2 Further, the cited references do not disclose or suggest said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods" that are executed in said block during the execution of said model as present in claim 1. For the reasons discussed above, none of the cited references disclose or suggest a proper or improper operation for any of plurality of block methods'." (Remarks: page 5 middle)

10. **Examiner Response:**

10.1 Regarding subsection 1 *supra*, the sole basis of the argument regarding ordering is based on Applicants' position that the cited art does not disclose the block methods as claimed. This was respectfully traversed above; thus, rendering the argument above moot.

10.2 Regarding subsection 2 *supra*, in view of MPEP 2111.04, which recites, "Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure," the feature of "allowing the user to determine..." is not given patentable weight. This is because the feature is drawn to intended use at best which does not necessitate function or structure. At best, it can be interpreted as an "ability" which is not patentable. Nevertheless, the mere creation of the block

methods "allows" a user to determine proper / improper operation at his or her discretion, but does not require him or her to do so.

11. **Applicants argue:**

11.1 "[Regarding claim 48] Alpern does not allow an execution method that operates in a first environment of a computer-based modeling application that executes a model to be debugged, and one having ordinary skill in the art would not have a reasonable expectation of success in implementing the operations of Alpern in a computer-based modeling application that executes a mode. Alpern is concerned with debugging modules for virtual machines that were written in different languages." (Remarks: page 8)

12. **Examiner Response:**

12.1 There is indeed expectation of success. A virtual machine is a model of an actual machine. Further, this argument is made on the basis of the argument presented for claim 1, which was traversed above.

13. **Applicants argue:**

13.1 "MathWorks and the Official Notice taken are concerned with block-level debugging. Instead of displaying a hierarchy of root and child methods, MathWorks displays a block execution order, and not information relating to root and child methods. For instance, compare MathWorks at page 12-16, showing the hierarchy of MathWorks, with Figure 18A of the Application, which depicts the parent-child hierarchy. The block execution order of MathWorks is silent as to root or child methods." (Remarks: page 13; emphasis in original)

14. **Examiner Response:**

14.1 The blocks displayed are displayed in a root-child relationship, and the blocks contain methods, as demonstrated above, the root-child block method relationship is also displayed inherently and implicitly. Accordingly, the argument is traversed.

15. All arguments have been addressed but not all have been specifically mentioned above, as they are repetitious of the core "block method" argument.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

16. Claims 1-3, 5-9, 17-22, 24, 25-27, 29-33, and 41-46, 48-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over MathWorks' Simulink "Dynamic System Simulation for MATLAB" "Using Simulink Version 2.2", 1997 (**"MathWorks"**), and further in view of **Official Notice taken** (admitted prior art) further in view of Alpern **(US 7107578)**.

As per claim 1, MathWorks discloses: In a modeling and execution environment, a method comprising the steps of:

providing a graphical debugger interfaces **(page 1-2 last paragraph)** with:

a model view of a model being executed **(page 1-2 last paragraph)**; and

a block method execution list view of block methods called during execution of said model **(page 3-49)**,

said model comprising a block having at least a method **(12-3; 9-37; 9-42 "The block can integrate using these methods: ...")**, said graphical debugger having debug information

related to the execution of said model **(12-3)**, and a start time or a stop time of said plurality of block methods for said block that are executed in said block during the execution of said model **(start time ... 12-3 last para; 2-12; stop time ... 4-2 "An important advantage is that you can perform certain operations interactively while a simulation is running: You can modify many simulation parameters, including the stop time, the solver, and the maximum step size.")**; and outputting said debug information to a user, said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of block methods that are executed during the execution of said model.

MathWorks implies but does not make explicit that "a block includes a plurality of block methods" **(12-3; 9-37; 9-42 "The block can integrate using these methods: ...")**.

Official Notice is taken with respect to this feature - Office Action is admitted prior art due to inadequate traversal.

The legal basis for the 35 U.S.C. § 103 rejection is detailed in MPEP 2144.04.VI.B titled "Duplication of Parts", wherein it is described that mere duplication of parts has no patentable significance unless a new and unexpected result is produced. In this instance, merely duplicating the number of methods that each block contains does not produce a new and unexpected result.

Motivation to do so would have been to create a more compact design, which is also not a patentably significant feature. See MPEP 2144.04.V.B.

Further, the combination of MathWorks and the Official Notice does not expressly disclose that "said debug information indicating an order of execution of said plurality of block methods for said block", however MathWorks hints at this **(12-16, 12-16 to 12-19, 12-5)**.

Alpern however discloses an analogous system having the said features **(Fig 5 and description; col: 12 line: 35-46: "The frame stack 530 of FIG. 5 is useful at a single debugger client because it shows the sequence of frames representing routines in the order executed.")**. Alpern further discloses the debug information having start and stop time of said plurality of block methods for said block **(col: 6 line: 29-43 with emphasis on lines 41-43; Fig 6C, 6D and**

description). An advantage of having detailed information such as the order of execution and the start / stop time of execution is that the programmer can better understand which parts of the program are unutilized / take up the longest amount of time. Therefore, the programmer could reduce that time and reducing the overall time by optimizing the sections of blocks (groups) / specific methods. Thereby saving computing time and costs associated therewith.

As per claim 2, Mathwork discloses: The method of claim 1, comprising the further steps of:

wrapping data generated by the execution of said model in an object, said wrapping encapsulating said execution-generated data in said object **(11-3: How to Specify a Path for a Simulink Object, 9-4 "To File", 9-61, -144, -145)**; and

exposing said data to said debugger via at least one interface to said object **(9-92 the exposure occurs when the debugger reads the information into the memory "From File", 9-61, -144, -145)**.

As per claim 3, MathWorks discloses: The method of claim 2, comprising the further step of: altering said data via said interface **(-131, 4-2: "An important advantage is that...")**.

As per claim 5, MathWorks discloses: The method of claim 1, comprising the further steps of:

processing said model to create compiled model information **(1-10 bullet 2, 1-12, 8-2: "C language S-functions are compiled as MEX-files using the mex utility described in the Application Program Interface Guide. As with other MEX-files, they are dynamically linked into MATLAB when needed.)**; and

programmatically generating executable code from said compiled model information, said code including an interface to said debugger **(1-12: linked, 8-36 first 3 para, 8-42: cg_sfun.h is included if the file is being used in conjunction with the Simulink Real-Time Workshop to produce a stand-alone or real-time executable.)**.

As per claim 6, MathWorks discloses: The method of claim 5, comprising the further step of:

Executing said generated code wherein said debugger at least one of sends or receives information from said executing code during said execution **(3-38: "Start Fcn: After the block**

diagram is compiled and before the simulation starts"; 12-2: "It [Simulink debugger] enables you to pinpoint problems by running simulations step-by-step and displaying intermediate block states and input and outputs.").

As per claim 7, MathWorks discloses: The method of claim 6, comprising the further steps of:

saving an execution history for said executable code **(MathWorks' "Target Language Compiler Reference Guide" ("TLC") futher expands on this inherent feature on page A-20 "This history is saved in the real-work vector.")**; and

outputting the execution history by at least one of saving it in a permanent memory location **(this feature is inherent)**, displaying it for a user **(the GUI displays the results to the users, furthermore, the data stored to the files is viewable by users)**, or sending it to a printing device to be printed **(RTW: 4-9, MathWorks: 3-26)**.

As per claim 8, MathWorks discloses: The method of claim 6 wherein said debugger is started after compilation and before the execution of said code **(this feature is inherent within the disclosure. Specifically, the debugger must have something to debug and therefore debugs after the compilation has finished. Furthermore, the debugger starts the execution of the code and is therefore started before the execution of the code.)**.

As per claim 9, MathWorks discloses: The method of claim 1, comprising the further step of:

indicating graphically using said debugger a plurality of blocks that are part of an algebraic loop when the executing model is processing the algebraic loop **(7-10, 12-14, 12-18, 4-20 first para)**.

As per claim 17, MathWorks discloses: The method of claim 1, comprising the further step of:

communicating with an external mode simulation with said debugger **(8-114: "SS_SIMMODE_EXTERNAL — External mode simulation")**.

As per claim 18, MathWorks discloses: The method of claim I, comprising the further step of:

saving a snapshot of data relating to model execution during execution of said model, said snapshot data sufficient to enable the subsequent restarting of the execution of said model using

said snapshot data **(4-16: "You can also save the final states for a simulation and apply them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state.")**.

As per claim 19, MathWorks discloses: The method of claim 18 wherein said snapshot data is saved programmatically at least one or more of a regular interval or based on a user-defined parameter **(4-16: "You can also save the final states for a simulation and apply them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state." The user defined parameter is whenever the user chooses to do so manually.)**.

As per claim 20, MathWorks discloses: The method of claim 19, comprising the further step of: loading a saved snapshot into said debugger; and

executing a saved model based on said saved snapshot, said saved model executing from a point in time said snapshot was saved using information from said saved snapshot **(4-16: "You can also save the final states for a simulation and apply them to another simulation. This feature might be useful when you want to save a steady-state solution and restart the simulation at that known state.")**.

As per claim 21, MathWorks discloses: The method of claim 18, comprising the further step of: displaying graphically to a user the saved snapshot data **(this feature is inherent when the snapshot is restarted)**.

As per claim 22, MathWorks discloses: The method of claim 21, comprising the further step of

displaying graphically to a user at least one additional set of snapshot data without restarting the execution of said model **(This feature is inherent, it is the filename of the snapshot.)**.

As per claim 24, MathWorks discloses: The method of claim 18, comprising the further step of:

saving a difference between a set of current model execution data and a saved snapshot **(this feature is inherent. Specifically, when the simulation is restarted from a snapshot point and later saved it will be saved with the difference incorporated within the new**

snapshot.).

As per claim(s) 25-27, 29-33, 41-46, note the rejection of claim(s) 1-3, 5-9, 17-22, 24 above. The Instant Claim(s) is/are functionally equivalent to the above-rejected claim(s) and is/are therefore rejected under same prior-art teachings.

As per claim 48, note the rejection of claims 1-2 above. The Instant Claim recites substantially same limitations as the above-rejected claims and therefore rejected under same prior-art teachings, but for: identifying a first execution method operating in a first environment of a modeling application that executes a model, where the first environment is one of a text-based environment, a time-based block diagram, a state based block diagram, or a data-flow diagram **(B-2 model file ... text-based environment);**

identifying a second execution method operating in a second domain, where the second domain differs from the first domain **(1-6: GUI-based tools for designing simulating and analyzing systems).**

As per claims 49-50, note the rejection of claim 1-2 above. The Instant Claims recite substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks discloses: 51. The method of claim 48, further comprising:

displaying a hierarchy containing information about the first execution method or the second execution method, the hierarchy allowing a user to identify relationships between the first execution method and the second execution method, the first execution method and another execution method, or the second execution method and the another execution method **(1-3: "You can view the system at a high-level, then double-click on blocks to go down through the levels to see increasing levels of model detail.")**.

MathWorks discloses: 53. The method of claim 48, further comprising: identifying the first execution method or the second execution method using a visual indicator to identify when the first execution method or the second execution method is executing **(12-5)**.

As per claim 52, note the rejection of claims 1-2, 51, 53 above. The Instant Claim recites substantially

same limitations as the above-rejected claims and therefore rejected under same prior-art teachings.

As per claim 54, note the rejection of claim 1 above. The Instant Claim recites substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks discloses: 55. (previously presented) A method, comprising:

identifying a first root method comprising one or more child methods, the first root method related to a graphical modeling application; identifying a second root method related to the graphical modeling application **(4-22: y, y1..yn)**;

running the first root method and the second root method in a graphical debugger to obtain information about the operation of the first root method or the second root method, the graphical debugger interfaced with a method list of methods called during simulation of a model in the graphical modeling application and a model view of a model in the graphical modeling application **(1-2; 3-49)**; and displaying a debugging result to a destination, the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method **(8-46, 8-111, 11-2; 3-51 (systems list); 4-25; 11-15; A-7; Alpern: Fig 5, 6A-D and descriptions)**.

As per claim 56, note the rejection of claim 55 above. The Instant Claim recites substantially same limitations as the above-rejected claim and therefore rejected under same prior-art teachings.

MathWorks discloses: 57. The method of claim 56, wherein the displaying an indicator further comprises:

displaying a first symbol when the status is related to the first root method; and displaying a second symbol when the status is related to the one or more child methods or the second root method **(3-19; 6-14; -118; 10-15; A-7)**.

MathWorks discloses: 58. The method of claim 56, wherein the displaying an indicator further comprises:

displaying a first color to represent a first status related to the first root method; and displaying a

second color to represent a second status related to one of the one or more child methods or the second root method **(3-19; 6-14; -118; 10-15; A-7)**.

MathWorks discloses: 59. The method of claim 56, further comprising:

displaying the hierarchy in a first region related to one or more display devices; and displaying a graphical diagram related to the first root method or the second root method in a second region related to the one or more display devices, the graphical diagram synchronized with information displayed in the first region **(3-19; 6-14; -118; 10-15; A-7)**.

As per claim 60-66, note the rejection of claims 50-51, 57-60 above. The Instant Claim recites substantially same limitations as the above-rejected claims and therefore rejected under same prior-art teachings.

MathWorks discloses: 66. The method of claim 65, wherein the first indicator or the second indicator are a color, a pointer, a symbol, a font, or a border **(A-7)**.

MathWorks discloses: 67. The method of claim 64, wherein the first display area comprises a window that displays information about the graphical icon or the graphical icon debugging information **(2-6; 2-7; 2-11; 3-49)**.

MathWorks discloses: 68. The method of claim 67, wherein the window comprises a visual indicator to connect the window to the graphical icon or to the graphical icon debugging information **(2-6; 2-7; 2-11; 3-49)**.

MathWorks discloses: 69. The method of claim 64, further comprising: displaying an execution list in the hierarchy, the execution list related to the root method or the one or more child methods **(3-49)**.

MathWorks discloses: 70. The method of claim 1, wherein the model comprises a plurality of blocks having execution methods, and wherein the debug information indicates an order of execution of said execution methods of said plurality of blocks, during execution of the model **(12-5: "The debugger executes the current block, stops, and highlights the next block in the model's block execution order (see "Displaying a Model's Block Execution Order" on page 12-16)."; 12-16)**.

MathWorks discloses: 71. (The medium of claim 25, wherein the model comprises a plurality of blocks having execution methods, and wherein the debug information indicates an order of execution of said execution methods of said plurality of blocks, during execution of the model **(12-5: "The debugger executes the current block, stops, and highlights the next block in the model's block execution order (see "Displaying a Model's Block Execution Order" on page 12-16)."; 12-16)**.

17. Claims 4, 11-16, 23, 28, 34-40, 47, and 72 are rejected under 35 U.S.C. 103(a) as being unpatentable over MathWorks's Simulink, 1997 ("MathWorks") as applied to claim 1 above, and further in view of **Official Notice taken** (admitted prior art) further in view of Alpern **(US 7107578)** and further in view of Fenlason's "GNU gprof" ("GNU gprof") (1998).

As per claim 4, MathWorks discloses all limitations of claim 1, and that the execution-generated data is at least one of state information **(4-16 "Loading and Saving States", -131, A-22: signal generators, etc, 8-65)**, block inputs, block outputs **(3-15, 8-46 "In general, block inputs and outputs are written", 9-80)**, solver data **(4-4, 4-6, 4-16)**, signal values for said model **(-119, 8-124)**. MathWorks however does not explicitly disclose profiling data. GNU gprof however discloses an analogous application profiling system having the said feature **(page 14, "The primary line of this entry describes the total time spent directly in the functions of the cycle.")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 72, MathWorks discloses: The method of claim 1, comprising the further step of: saving a record of a unique execution method invocation, **(1-3: "After you define a model, you can simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in MATLAB's command window.")**. MathWorks however does not substantially disclose said execution unique execution method invocation comprising information related to the execution of

one of said plurality of execution methods that belongs to said block or to another block in the model, a system, or a model instance in an execution list of called execution methods. GNU gprof however discloses an analogous application profiling system having the said feature **(page 11: Call Graph)**. Further, the action of displaying / outputting the call graph is functionally equivalent to saving temporarily (while the display is On and the screen is not changed). The motivation for this combination would have been the same as recited above.

As per claim 11, MathWorks discloses all limitations of claim 10. MathWorks does not expressly disclose that the unique execution method invocation record comprises information about child records of a subset of said plurality of execution executed inside said unique execution method invocation record. GNU gprof however discloses the said features **(page 12 section titled "children")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 12, MathWorks discloses all limitations of claim 11. MathWorks however does not expressly disclose that a link is provided from said unique execution method invocation record to said child record. GNU gprof however discloses an analogous system having the said feature **(page 6 section titled "--file-ordering map_file": "The `--file-ordering' option causes gprof to print a suggested .o link line ordering for the program based on profiling data.")**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 13, MathWorks discloses all limitations of claim 10. MathWorks does not however expressly disclose that the said unique execution method invocation record comprises information regarding at least one parent record of one or more of the plurality of execution methods in which said unique execution method invocation is executed. GNU gprof however discloses an analogous system having the said feature **(page 11: Call Graph)**. It would have been obvious to one of ordinary skill in the art at the

time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 14, MathWorks discloses all limitations of claim 13. MathWorks however does not expressly disclose a link is provided from said unique execution method invocation record to said parent record. GNU gprof however discloses an analogous system having the said feature **(page 6 section titled "--file-ordering map_file": "The `--file-ordering' option causes gprof to print a suggested .o link line ordering for the program, page 11: Call Graph)**. It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the references in order to time the execution of a program and routines of the program in order to identify which portions of the program cause a bottleneck and resolve them.

As per claim 15, MathWorks discloses all limitations of claim 72. MathWorks however does not expressly disclose that the said unique execution method invocation record comprises data about a state of an invocation of block method. GNU gprof however discloses an analogous system having the said feature **(page 11: Call Graph - called column)**.

As per claim 16, MathWorks discloses all limitations of claim 15. MathWorks however does not expressly disclose that the said state indicates the method invocation is at one of the states of entering, entered, exiting and exited **(page 11: Call Graph)**.

As per claim 23, MathWorks discloses all limitations of claim 22. MathWorks however does not expressly disclose that the said set of snapshot data is displayed in order of decreasing time. This is merely a design choice. Microsoft Windows allows for sort of descending or ascending names, file types, sizes, creation and modification dates. This is done for faster searching and identification of the user-required information.

As per claim(s) 28, 34-40, and 47, note the rejection of claim(s) 4, 72, 11-16, and 23 above. The Instant Claim(s) is/are functionally equivalent to the above-rejected claim(s) and is/are therefore rejected under same prior-art teachings.

Support for Amendments and Newly Added Claims

Applicants are respectfully requested, in the event of an amendment to claims or submission of new claims, that such claims and their limitations be directly mapped to the specification, which provides support for the subject matter. This will assist in expediting compact prosecution. MPEP 714.02 recites: "Applicant should also specifically point out the support for any amendments made to the disclosure. See MPEP § 2163.06. An amendment which does not comply with the provisions of 37 CFR 1.121(b), (c), (d), and (h) may be held not fully responsive. See MPEP § 714." **Amendments not pointing to specific support in the disclosure may be deemed as not complying with provisions of 37 C.F.R. 1.131(b), (c), (d), and (h) and therefore held not fully responsive.** Generic statements such as "Applicants believe no new matter has been introduced" may be deemed insufficient.

Conclusion

18. All claims are rejected.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Silver whose telephone number is (571) 272-8634. The examiner can normally be reached on Monday thru Friday, 10am to 6:30pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on 571-272-2279. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/David Silver/

Examiner, Art Unit 2128